63-3-2

THE UNIVERSITY OF WISCONSIN

madison, wisconsin

UNITED STATES ARMY

MATHEMATICS RESEARCH CENTER

MATHEMATICS RESEARCH CENTER, UNITED STATES ARMY
THE UNIVERSITY OF WISCONSIN

Contract No. : DA-11-022-ORD-2059

ON CONTINUED FRACTION EXPANSIONS FOR BINOMIAL QUADRATIC SURDS — — COMPUTATION ON COMPUTERS

E. Frank

MRC Technical Summary Report #363
March 1963

Madison, Wisconsin

ON CONTINUED FRACTION EXPANSIONS FOR

BINOMIAL QUADRATIC SURDS — —

COMPUTATION ON COMPUTERS

E. Frank

1. Introduction

In [1], [2],* the author studied the regular continued fraction expansion

$$(1.1) \qquad [b_0, \overline{b_1, b_2, \dots, b_p}]$$

for the binomial quadratic surd $\sqrt{C} + L$, where C and L are rational numbers, C positive. Simultaneously, the approximations x_n to $\sqrt{C} + L$ given by an extension of Newton's formula,

$$(1.2) \qquad x_n = \frac{x_i \cdot x_j + C - L^2}{x_i + x_j - 2L} ,$$

where x_i and x_j are certain previous approximations to the value of $\sqrt{C}+L$, were studied. Let $A_i/B_i = x_i$ denote the i-th approximant of (1.1). Then in [1], [2], a complete classification was derived concerning which ones of the A_i/B_i are also approximations to $\sqrt{C} + L$ given by formula (1.2).

---

* Numbers in brackets refer to the bibliography at the end of the paper.

Sponsored in part by the Technische Hochschule, München, Germany, and in part by the Mathematics Research Center, U. S. Army, Madison, Wisconsin, under Contract No. DA-11-022-ORD-2059.

In the present study, the use of automatic digital computers is demonstrated for the rapid computation of the regular continued fraction expansion (1.1) for $\sqrt{c} + L$, of the approximants of (1.1), and of the approximations by formula (1.2). The use of an electronic computer is especially convenient here since the same operations are repeated many times for many combinations of the various values of the parameters.

Simultaneously, the aim of this paper is a demonstration of the use of computers in problems in numerical analysis, and, in particular, in their application to continued fractions and to number theory. Very few such applications have thus far been made. Consequently, they are important since they point the way in which extensions are possible in applications of continued fraction theory in functional analysis, and in approximations of functions by rational fractions through the use of continued fractions.

In §2 are given the mathematical description of the problem to be solved and the formulas and methods by which it is solved. A summary of the calculation procedure is given in §3. It contains an Algol program for the algorithms involved (Table I) and a sample problem computed on the PERM machine (Technische Hochschule, München) (Table II). Of interest for comparison purposes, Table III contains a flow chart and description thereof for the Fortran program of the previous algorithms (Table IV). Table V shows the preceding problem worked out with the Fortran program and the Control Data Corporation 1604 machine. It will be noted that the Fortran program is ready to put directly into the machine. On the other hand, the Algol program must be incorporated into a main program.

2. Mathematical description

I. The first problem is the derivation of formulas for the computation of the numbers b_i in the regular periodic continued fraction expansion for $\sqrt{C} + L$, namely,

$$(2.1) \qquad b_0 + \frac{1}{b_1} + \frac{1}{b_2} + \ldots + \frac{1}{b_p} + \frac{1}{b_1} + \frac{1}{b_2} + \ldots,$$

(formula (1.1)). Lagrange showed that this continued fraction is always periodic. It is also symmetric if $L = 0$ (cf., for example, [3]).

One is given $\sqrt{C} + L$. This can be written $\dfrac{\sqrt{D} + P_o}{Q_o}$,

where $(D - P_o^2)$ is divisible by Q_o. Then b_o is the largest integer contained in $\dfrac{\sqrt{D} + P_o}{Q_o}$. Thus, one is given the values of D, P_o, Q_o, and b_o. The continued fraction (2.1) is generated by a sequence of linear transformations of the type

$$(2.2) \qquad f_v = b_v + \frac{1}{fv + 1}, \quad v = 0, 1, \ldots .$$

Now,

$$\frac{\sqrt{D}+P_v}{Q_v} = b_v + \frac{Q_{v+1}}{\sqrt{D}+P_{v+1}}, \tag{2.3}$$

or

$$D + P_vP_{v+1} + (P_v + P_{v+1})\sqrt{D} = b_vQ_v\sqrt{D} + b_vQ_vP_{v+1} + Q_vQ_{v+1} .$$

Since $\sqrt{D}$ is irrational,

$$D + P_vP_{v+1} = b_vQ_vP_{v+1} + Q_vQ_{v+1}, \tag{2.4}$$

$$P_v + P_{v+1} = b_vQ_v . \tag{2.5}$$

When one multiplies equation (2.5) by P_{v+1} and subtracts the result from (2.4), one obtains

$$D - P_{v+1}^2 = Q_vQ_{v+1}, \; v = 0, 1, \ldots .$$

Then, when one replaces v by $v - 1$ in the last equation and subtracts the result from (2.5), one obtains

$$Q_vQ_{v+1} - Q_{v-1}Q_v = P_v^2 - P_{v+1}^2 = (P_v - P_{v+1})(P_v + P_{v+1}) = (P_v - P_{v+1})\, b_vQ_v ,$$

or

$$Q_{v+1} - Q_{v-1} = b_v\,(P_v - P_{v+1}). \tag{2.6}$$

From (2.5) and (2.6) one obtains the recurrence relations

$$(2.7)\qquad \begin{aligned} &P_{v+1} = b_v Q_v - P_v , \\ &Q_{v+1} = b_v (P_v - P_{v+1}) + Q_{v-1} , \ v = 0, 1, \ldots , Q_{-1} = \frac{D - P_o^2}{Q_o} . \end{aligned}$$

Also

$$(2.8)\qquad b_v = \text{largest integer contained in } \frac{\sqrt{D} + P_v}{Q_v} = \left[\frac{\sqrt{D} + P_v}{Q_v}\right] .$$

Formulas (2.7) and (2.8) are those required for the computation of the P_v, Q_v, and b_v, $v = 1, 2, \ldots$. In addition, one uses the recurrence formulas for the A_v and B_v, the numerator and denominator of the v-th approximant of (2.1):

$$(2.9)\qquad \begin{aligned} &A_v = b_v A_{v-1} + A_{v-2} , \ B_v = b_v B_{v-1} + B_{v-2} , \ v = 1, 2, \ldots , \\ &A_{-1} = 1 , \ A_o = b_o , \ B_{-1} = 0 , \ B_o = 1 . \end{aligned}$$

It is remarked that, if $\sqrt{C} + L$ is negative, one writes $-(-\sqrt{C} - L)$ and carries out the above computation for $-\sqrt{C} - L$. The minus sign is then later added to the continued fraction, namely,

$$= [b_o, \overline{b_1, b_2, \ldots, b_p}] \quad .$$

Furthermore, it is understood that $\sqrt{C}$ is the positive root, since, if one is given $\frac{-\sqrt{D}+P_o}{Q_o}$, one writes the equivalent $\frac{\sqrt{D}-P_o}{-Q_o}$.

Also the computation involves only integers if one first multiplies numerator and denominator of the binomial quadratic surd in question by such a constant that $\frac{D - P_o^2}{Q_o} = Q_{-1}$ is an integer.

In order to find the period p, one notes that one has obtained a complete period when the values of P_i, Q_i, and b_i are repeated. In the following example (Tables II and V), these values are compared with P_1, Q_1, and b_1 since there is only one initial term b_o. However, if the periodic part starts with b_j, $j = 2, 3, \ldots$, one uses as comparison numbers P_j, Q_j, and b_j.

II. Formula (1.2) is simply an estimate of the roots of $x^2 - 2Lx - (C - L^2) = 0$. It is an extension of Newton's formula. In [1] the author showed that, if x_{kp-s} is the [(kp - s)]-th approximant of (1.1), then

$$x_{kp-s} = \frac{x_{(k-r)p-s} \cdot x_{rp-1} + C - L^2}{x_{(k-r)p-s} + x_{rp-1} - 2L} \quad , \tag{2.10}$$

$k = 2, 3, \ldots\ r = 1, 2, \ldots\ k - r \geq 1,\ p = 1, 2, \ldots,$

where $x_{(k-r)p-s}$ and x_{rp-1} are the $[(k-r)p-s]$-th and $(rp-1)$-th approximants, respectively, of (1.1). Sharper formulas for x_{kp-s}, in terms of i, were obtained in [1] and [2] when (1.1) is symmetric of even ($p = 2i$) and odd ($p = 2i+1$) period. However, for illustrative purposes in this study, formula (2.10) is used.

In order to use formula (2.10) on the machine to compute A_{kp-s} and B_{kp-s}, one must write the formulas for these two quantities in either one of the two following ways (by means of the formulas given in [1]):

$$(2.11)\quad \begin{aligned} A_{kp-s} &= A_{(k-r)p-s}A_{rp-1} + [(b_p - b_o)A_{rp-1} + A_{rp-2}]B_{(k-r)p-s}, \\ B_{kp-s} &= A_{(k-r)p-s}B_{rp-1} + [(b_p - b_o)B_{rp-1} + B_{rp-2}]B_{(k-r)p-s}; \end{aligned}$$

$$(2.12)\quad \begin{aligned} Q_o^2 A_{kp-s} &= Q_o^2 A_{(k-r)p-s}A_{rp-1} + (D - P_o^2)B_{(k-r)p-s}B_{rp-1}, \\ Q_o^2 B_{kp-s} &= Q_o^2 (A_{(k-r)p-s}B_{rp-1} + A_{rp-1}B_{(k-r)p-s}) - 2P_oQ_oB_{(k-r)p-s}B_{rp-1}. \end{aligned}$$

Similar transformations are necessary for the other formulas given in [1] and [2].

III. As an estimate of the error involved, the formula (cf. (1.3) of [1])

$$\left| (\sqrt{C} + L) - \frac{A_i}{B_i} \right| < \frac{1}{B_i^2}, \quad i = 1, 2, \dots, \tag{2.13}$$

is used here.

3. Algol and Fortran Programs and a Sample Problem.

TABLE I

ALGOL PROGRAM

```
'PROCEDURE' CONTINUED FRACTION( D, Po, Qo, Co, N) RESULT: (A, B, C, E, PERIOD);
   'INTEGER' D, Po, Qo, Co, N, PERIOD;'INTEGER' 'ARRAY' A, B, C; 'ARRAY' E;
   'BEGIN' 'INTEGER' NY;  'INTEGER' 'ARRAY' P[o: N],  Q[-1: N];

               P[o] : = Po;
               Q[o] : = Qo;
               A[o] : = C[o] : = Co;
               A[-1] : = B[o] : = E[o] : =1;
               B[-1] : = PERIOD : = o;
               Q[-1] : = (D - PoxPo)/Qo;
```

```
'FOR' NY : = 1 'STEP' 1 'UNTIL' N 'DO'
  'BEGIN'
     P[NY] : = C[NY - 1] xQ [NY - 1] - P[NY - 1] ;
     Q[NY] : = C[NY - 1] x(P[NY - 1] - P[NY]) + Q[NY - 2];
     C[NY] : ENTIER((SQRT(D) + P[NY])/Q[NY]) ;
     A[NY] : = C[NY] xA[NY-1] + A[NY - 2] ;
     B[NY] : = C[NY] xB [NY - 1] + B[NY - 2] ;
     E[NY] : = 1/(B[NY] xB[NY]) ;
     'IF' NY 'GREATER' 1 'AND' (P[NY] 'EQUAL' P[1] 'AND'
     Q[NY] 'EQUAL' Q[1]) 'AND' PERIOD 'EQUAL' o 'THEN'
     PERIOD : = NY - 1
'END'
'END' CONTINUED FRACTION;

'PROCEDURE' EXTENSION(N, M, A, B, C, E, PERIOD) ;
   'INTEGER' N, M, PERIOD; 'INTEGER' 'ARRAY' A,B,C; 'ARRAY' E ;
 'BEGIN' 'INTEGER' K,S, I, NY;
   'IF' PERIOD 'EQUAL' o 'THEN'
     'BEGIN' WRITE ( " NO PERIOD " ) ; 'GOTO' EXIT EXTENSION 'END' ;
   'FOR' NY : = N + 1 'STEP' 1 'UNTIL' M 'DO'
```

```
'BEGIN'
      K : = ENTIER( NY/PERIOD) + 1 ;
      S : = KxPERIOD-NY;
      1 : = (K-1)xPERIOD-S;
      A[ NY] : = A[ 1]xA [ PERIOD - 1] +( (C[ PERIOD]-C[ o]) xA[ PERIOD - 1] +
             A[ PERIOD - 2]) xB[ 1] ;
      B[ NY] : = A[ 1] xB[ PERIOD - 1] + ((C[ PERIOD] - C[ o]) xB[PERIOD - 1] +
             B[ PERIOD - 2]) xB[ 1] ;
      E[ NY] : = 1/( B[ NY] xB[ NY])

'END' ;
EXIT EXTENSION : 'END' EXTENSION ;

'PROCEDURE' CORRECT FORMULA( D, Po, Qo, M, L, A, B, C, E, PERIOD) ;
   'INTEGER' D, Po, Qo, M, L, PERIOD; 'INTEGER' 'ARRAY' A, B, C; 'ARRAY' E;
  'BEGIN' 'INTEGER' K, S, I, NY;
   'IF' PERIOD 'EQUAL' o 'THEN'
     'BEGIN' WRITE ('' NO PERIOD '' ); 'GOTO' EXIT CORRECT FORMULA
         'END' ;

   'FOR' NY : = M + 1 'STEP' 1 'UNTIL' L 'DO'

     'BEGIN'
```

```
K : = ENTIER(NY/PERIOD) + 1;

S : = KxPERIOD - NY;

I : = (K - 1)xPERIOD - S;

A[NY]:=A[1]xA[PERIOD-1]+((D-PoxPo)xB[1]xB[PERIOD-1])/(QoxQo);

B[NY] : = A[1]xB[PERIOD-1] + A[PERIOD - 1]xB[1] -(2xPoxB[1]
        xB[PERIOD - 1])/Qo;

E[NY] : = 1/(B[NY]xB[NY])

'END';

EXIT CORRECT FORMULA : 'END' CORRECT FORMULA;
```

Here $b = c$ and $r = 1$.

TABLE II

Algol Program and PERM Computer

$$\frac{\sqrt{96} + 10}{4}$$

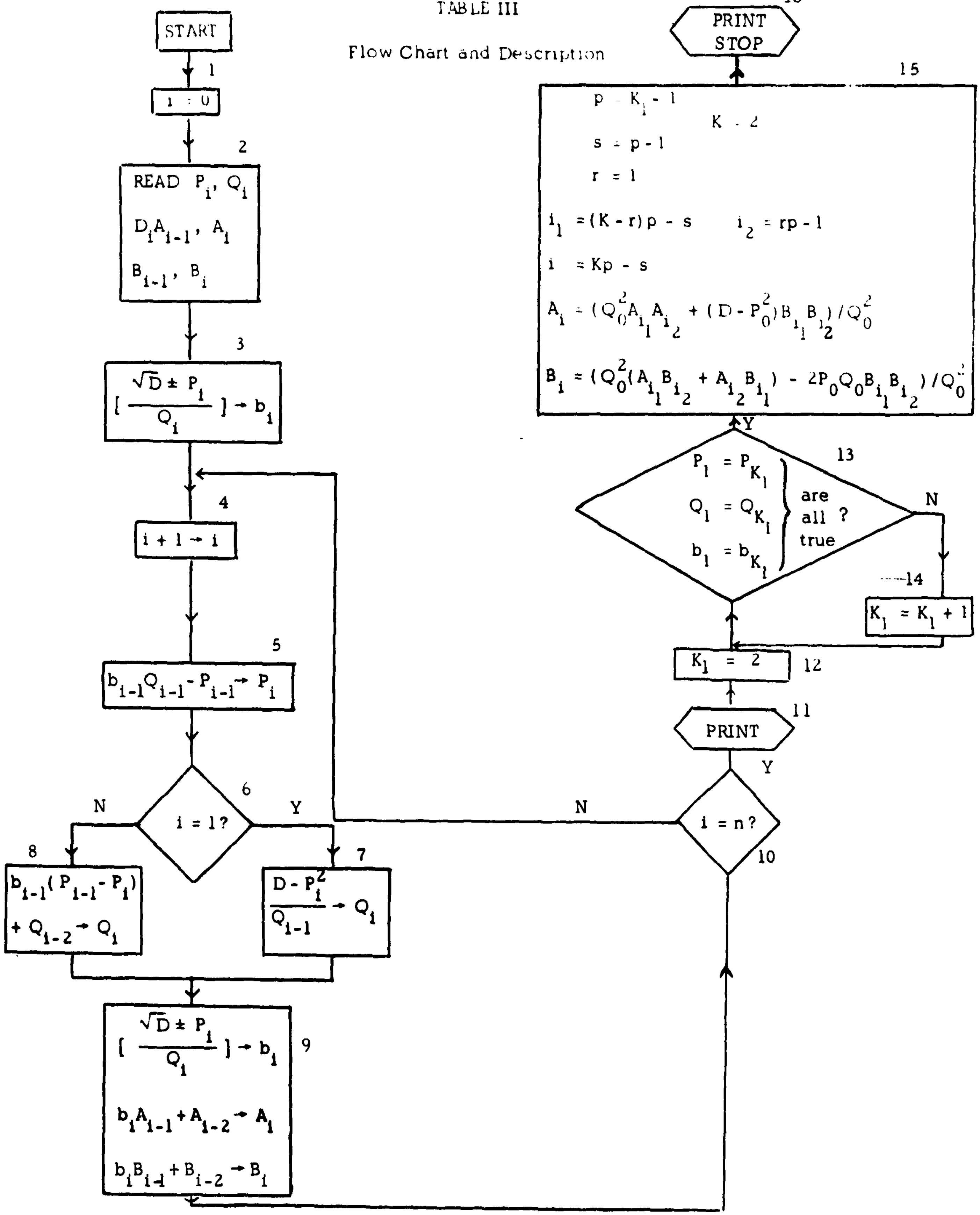
INPUTS:

D: 96 Po: 10 Qo: 4 Co: 4

SUBSCRIPT	A	B	C	1/(BxB)
RESULTS OF PROCEDURE CONTINUED FRACTION:				
0	4	1	4	+1.00000000
1	5	1	1	+1.00000000
2	94	19	18	+0.00277008
3	99	20	1	+0.00250000
4	391	79	3	+0.00016023
5	490	99	1	+0.00010203
6	9211	1861	18	+0.00000029
7	9701	1960	1	+0.00000026
8	38314	7741	3	+0.00000002
9	48015	9701	1	+0.00000001
10	902584	182359	18	+0.00000000
RESULTS OF PROCEDURE EXTENSION:				
11	950599	192060		+0.00000000
12	3754381	758539		+0.00000000
RESULTS OF PROCEDURE CORRECT FORMULA:				
13	4704980	950599		+0.00000000
14	88444021	17869321		+0.00000000
15	93149001	18819920		+0.00000000

TABLE III

Flow Chart and Description



Description of the Flow Chart

Box

1 The index i is set equal to zero.

2 The quantities P_i, Q_i, D, A_{i-1}, A_i, B_{i-1}, B_i are read as input.

3 The quantity $\left[\frac{\sqrt{D} \pm P_i}{Q_i}\right]$ is computed and designated as b_i.

4 The index i is increased by 1.

5 The quantity $b_{i-1}Q_{i-1} - P_{i-1}$ is computed and from here on is designated as P_i.

6 If $i = 1$, box 7 is carried out; if $i \neq 1$, box 8 is carried out.

7 The quantity $\frac{D - P_i^2}{Q_{i-1}}$ is computed and from here on is designated as Q_i.

8 The quantity $b_{i-1}(P_{i-1} - P_i) + Q_{i-2}$ is computed and from here on is designated as Q_i.

9 a) The quantity $\left[\frac{\sqrt{D} \pm P_i}{Q_i}\right]$ is computed and from here on is designated as b_i.

b) The quantity $b_i A_{i-1} + A_{i-2}$ is computed and from here on is designated as A_i.

c) The quantity $b_i B_{i-1} + B_{i-2}$ is computed and from here on is designated as B_i.

10 If $i = n$, the instruction in box 11 is carried out; if $i \neq n$, the computation begins again with box 4.

11 The results are printed.

12 The value K_1 is set equal to 2.

13 If $P_1 = P_{K_1}$, $Q_1 = Q_{K_1}$, and $b_1 = b_{K_1}$, the computation proceeds to box 15; otherwise, the computation proceeds to box 14.

14 The quantity K_1 is increased by 1.

15 a) The quantity $K_1 - 1$ is computed and from here on is designated as p.

b) The quantity $p - 1$ is computed and from here on is designated as s.

c) The quantity r is set equal to 1.

d) The quantity K is set equal to 2.

e) The quantity $(K - r)p - s$ is computed and from here on is designated as i_1.

f) The quantity $rp - 1$ is computed and from here on is designated as i_2.

g) The quantity $Kp - s$ is computed and from here on is designated as i.

h) The quantity $(Q_o^2 (A_{i_1} A_{i_2}) + (D - P_o^2) B_{i_1} B_{i_2})/Q_o^2$ is computed and from here on is designated as A_i.

i) The quantity $(Q_o^2 (A_{i_1} B_{i_2} + A_{i_2} B_{i_1}) - 2P_o Q_o B_{i_1} B_{i_2})/Q_o^2$ is computed and from here on is designated as B_i.

16 The results are printed and the computation ended.

TABLE IV

FORTRAN PROGRAM

```
      THE EXPRESSION OF A QUADRATIC SURD AS A CONTINUED FRACTION.
      PROGRAM DONE ON CDC 1604
      DIMENSION LP(50) , LQ(50) , LB(50) , LA(50) , LC(50)
      LLL = 5
      IA = 14073000
      IA = IA*10000000
    1 READ 2, LPZ, LQZ, LD, N
    2 FORMAT (3I10, I4)
  200 FORMAT ( 1H  I6 , I16, 4I21)
      PRINT 3, LPZ, LQZ, LD, N
    3 FORMAT ( 4H1PO = I10, 8H       QO = I10, 7H       D = I10, 7H       N = I4)
      M = SQRTF (FLOATF(LD))
    5 LCZ = (LPZ + M)/LQZ
    7 LP (1) = LCZ*LQZ - LPZ
      LQ (1) = (LD - LP(1) *LP(1))/LQZ
      IF (LQ(1))10, 28, 10
   10 LC (1) = ( M + LP(1))/LQ(1)
   11 LA(1) = LC (1) * LCZ + 1
      LAZ = LCZ
      LBZ = 1
      LB (1) = LC (1)
      LP( 2) = LC (1) * LQ(1) - LP(1)
```

```
      LQ(2) = LC(1)*(LP(1) - LP(2)) + LQZ
      IF(LQ(2)) 14, 28, 14
14    LC(2) = (M + LP(2))/LQ(2)
15    LA(2) = LC(2)*LA(1) + LAZ
      LB(2) = LC(2)*LB(1) + LBZ
      I = 0
      PRINT 100
100   FORMAT (116H    SUBSCRIPT              P                 Q
     1             C                A                  B            )
      PRINT 200, I, LPZ, LQZ, LCZ, LAZ, LBZ
      IB = 1
      PRINT 200, IB, LP(1), LQ(1), LC(1), LA(1), LB(1)
      IC = 2
      PRINT 200, IC, LP(2), LQ(2), LC(2), LA(2), LB(2)
      DO 22 J = 3, N
      LP(J) = LC(J - 1)*LQ(J - 1) - LP(J - 1)
      LQ(J) = LC(J - 1)*(LP(J - 1) - LP(J)) + LQ(J - 2)
      IF(LQ(J)) 19, 28, 19
19    LC(J) = (M + LP(J))/LQ(J)
20    LA(J) = LC(J)*LA(J - 1) + LA(J - 2)
      IF(LA(J) - IA) 21, 21, 29
21    LB(J) = LC(J)*LB(J - 1) + LB(J - 2)
      IF(LB(J) - IA) 30, 30, 31
30    IC = IC + 1
      PRINT 200, IC, LP(J), LQ(J), LC(J), LA(J), LB(J)
```

```
 22   CONTINUE

 28   PRINT 800
800   FORMAT(7H0Q(J) = 0)
      GO TO  190
 29   PRINT  900
900   FORMAT(14H OVERFLOW ON A)
      GO TO  190
 31   PRINT  901
901   FORMAT (14H OVERFLOW ON B)
      GO TO 190
190   PRINT  191
191   FORMAT(43H  MORE  A  AND  B  USING NEWTON EXTENDED FORMULA)
      PRINT  192
192   FORMAT(31H  SUBSCRIPT               A                B)
      KK = 2
599   IF(LP(1) - LP(KK)) 600, 601, 600
600   KK = KK + 1
      GO TO 599
601   IF(LQ(1) - LQ(KK)) 602, 603, 602
602   KK = KK + 1
      GO TO 599
603   IF(LC(1) - LC(KK)) 604, 605, 604
604   KK = KK + 1
      GO TO 599
```

```
605     MP = KK - 1
        MS = MP -1
        IF( MS) 505,501,505
505     MR = 1
        MK = 2
        DO  501  IS = 1 ,  MS
        I1  = ( MK  -  MR) *  MP  -  IS
        I2  =  MR *  MP  -  1
        II  =  MK *  MP  -  IS
        LA(II) = (LQZ**2)*LA(I1 )*LA( I2)+ ( LD - LPZ**2)*LB( I1 )*LB( I2 )
        LB(II) =LQZ**2*(LA(I1 )*LB( I2) + LA( I2) *LB( I 1)) - 2*LPZ*LQZ*LB( I1 )*LB(I2
       1)
        LA( II)  =  LA( II) /LQZ**2
        LB( II)  =  LB( II) /LQZ**2
        PRINT  333,  II,  LA( II),  LB(II)
333     FORMAT( I8,  2I16)
501     CONTINUE
        IF( LLL) 911,  700,  700
700     LLL = LLL  -  1
        GO TO  1
911     STOP
        END
        END
```

TABLE V

Fortran Program and Control Data 1604

PO = 10 QO = 4 D = 96 N = 30

SUBSCRIPT	P	Q	C	A	B
0	10	4	4	4	1
1	6	15	1	5	1
2	9	1	18	94	19
3	9	15	1	99	20
4	6	4	3	391	79
5	6	15	1	490	99
6	9	1	18	9211	1861
7	9	15	1	9701	1960
8	6	4	3	38314	7741
9	6	15	1	48015	9701
10	9	1	18	902584	182359
11	9	15	1	950599	192060
12	6	4	3	3754381	758539
13	6	15	1	4704980	950599
14	9	1	18	88444021	17869321
15	9	15	1	93149001	18819920
16	6	4	3	367891024	74329081
17	6	15	1	461040025	93149001
18	9	1	18	8666611474	1751011099
19	9	15	1	9127651499	1844160100
20	6	4	3	36049565971	7283491399
21	6	15	1	45177217470	9127651499
22	9	1	18	849239480431	171581218381
23	9	15	1	894416697901	180708869880
24	6	4	3	3532489574134	713707828021
25	6	15	1	4426906272035	894416697901
26	9	1	18	83216802470764	16813208390239
27	9	15	1	87643708742799	17707625088140
28	6	4	3	64672951988506	69936083654659

OVERFLOW ON A

MORE A AND B USING NEWTON EXTENDED FORMULA

SUBSCRIPT	A	B
7	9701	1960
6	9211	1861
5	490	99

The author wishes to thank Dr. Urich and Mr. R. Zirngibl of the Technische Hochschule, Müchen, and Mr. Jaafar Al-Abdulla of the Mathematics Research Center, U. S. Army, Madison, Wisconsin for help in setting up the programs in this paper.

BIBLIOGRAPHY

1. E. Frank, On Continued Fraction Expansions for Binomial Quadratic Surds I, Numerische Mathematik, v. 4, 85-95(1962).

2. E. Frank, On Continued Fraction Expansions for Binomial Quadratic Surds II, Numerische Mathematik, v. 4, 303-309(1962).

3. O. Perron, Die Lehre von den Kettenbrüchen, Vol. 1, Teubner, Stuttgart, (1954).